



CASTOR status and plans: proposals for SC3

5/4/2005



Outline



- CASTOR status & plans
- SC3 proposal
 - CASTOR version
 - SRM version



CASTOR status & plans



- The CASTOR version running in production follows the original CASTOR design defined in 1999 – 2000 and fully deployed for production use in early 2002
 - Name space in Oracle, MySQL → good scalability (currently >33M files)
- Known limitations:
 - The stager disk residence catalog is in memory → scales to O(200k) resident files but not beyond
 - No file access request throttling
 - No controlled resource sharing
- Stager limitations has lead to a deployment with many stager instances
 - ~40 instances, each with each its own disk pool
 - Bad sharing and loadbalancing
 - Difficult to operate



CASTOR status & plans



- New stager (disk pool manager)
 - Disk file residence catalog and request repository in Oracle (later also MySQL) → good scalability, stateless daemons
 - Externalized request scheduling (Maui or LSF supported) → throttling and controlled resource sharing
 - Many features targeted for easy operation and automated management
- New stager status:
 - Developments started in 2003
 - First version of complete new system was ready for testing in late 2004
 - Running in testbed since 4 months
 - Functionality OK
 - Performance:
 - Tier-0 requirements heavily tested with satisfactory results
 - Tier-1 requirements tests still ongoing. Request handling requires more tuning
 - New stager API not compatible with old stager → new SRM required



New stager



- Not going to present the architecture here. Please refer to CASTOR presentations presented at various occasions in 2004-2005, e.g.
 - **IEEE MSST 2004**
 - <http://cern.ch/castor/PRESENTATIONS/2004/MSST2004/MSST2004-40.html>
 - <http://cern.ch/castor/PRESENTATIONS/2004/MSST2004/MSST2004-CASTOR-1.pdf>
 - **CHEP04**
 - <http://cern.ch/castor/PRESENTATIONS/2004/chep04/index.html>
 - **Operational experience**
 - <http://cern.ch/castor/PRESENTATIONS/2005/CCM-20050118/NewStager.htm>
 - <http://cern.ch/castor/PRESENTATIONS/2005/CCM-20050322/ALICE-experience.html>
 - **API documentation**
 - <http://cern.ch/castor/DOCUMENTATION/CODE/STAGE/NewAPI/index.html>



CASTOR status & plans



- New stager deployment
 - Roll-out plan is being prepared
 - It is too early to give a definite timescale
 - Gain operational experience with test instances
 - Configuration, monitoring, automated management
 - Understand the Oracle requirements
 - Hardware, tuning, administration
 - User interfaces may require more discussions with experiments
 - RFIO backward compatible
 - Stager commands are only partially backward compatible
 - Stager API is completely new
(<http://cern.ch/castor/DOCUMENTATION/CODE/STAGE/NewAPI/index.html>)
 - Work out a good deployment architecture exploiting the advantages of the new stager
 - Scalable catalog and resource sharing facilities → fewer instances



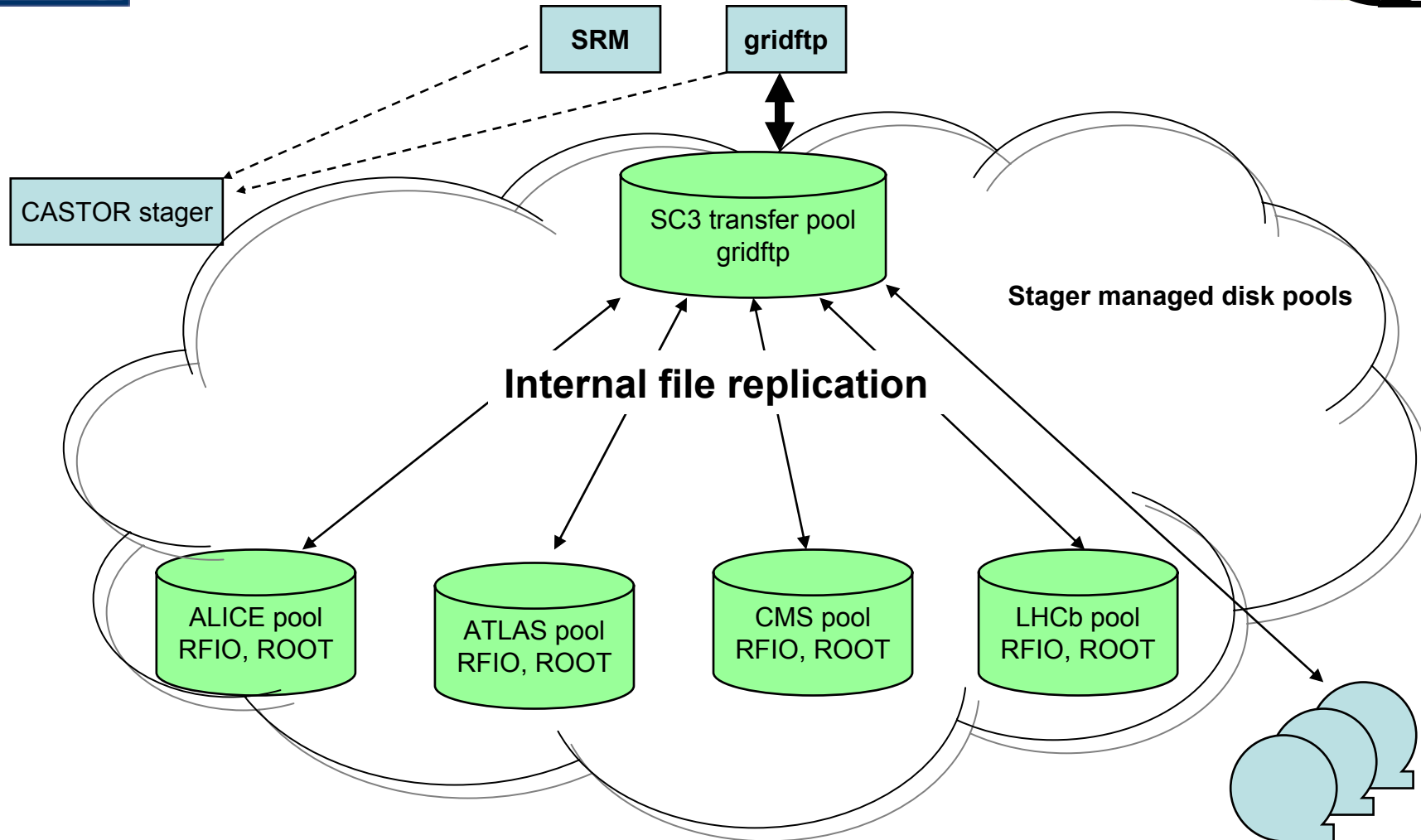
SC3 proposal



- **CASTOR version:**
 - **Throughput phase**
 - Use the new stager
 - **Service phase**
 - Can use new stager, but...
 - Sharing data with production disk pools only possible if the participating experiments have been migrated
 - New stager cannot share disk pools with an old stager instance



Possible SC3 deployment using new stager





SC3 proposal



- SRM version

- Strongly recommend to stay with V1.1

- It works and is well tested and known to interoperate but it took a long time and a lot of efforts to get there
 - Enough functionality for the throughput phase

- Going for a newer version is associated with high risks

- Will not interoperate out-of-the-box!
 - True (and respected) space reservations is well understood, e.g. what does 1TB reserved space mean?
 - I can write a 1TB file without hitting ENOSPC
 - I can write 1000 x 1GB files, each without hitting ENOSPC
 - V2.1 definition does not address this problem
 - The operational nightmare: "ls"
 - You would better gridftp the SOAP message resulting from srmLs on some castor directories



SRM implementations



- With the new stager, CASTOR needs a new SRM layer
- Trying re-using somebody else's framework
 - RAL offered to wrap the new CASTOR stager API in their SRM framework
 - Being tested this week
 - Other SRM 'providers' wanting to bid?
 - Everybody using the same SRM layer would be wonderful!
 - Spec 'gaps' handled in a single place
 - GSI interoperability guaranteed
 - But... operational support is more vital than the beauty of the implementation